

Clase 2 — 24.10.25

#VSC

Desarrollo de interfaces. JAVASCRIPT, JQUERY, REALIDAD VIRTUAL

Clase 2 — 24/10/2025

Tema: Primeros pasos con Javascript

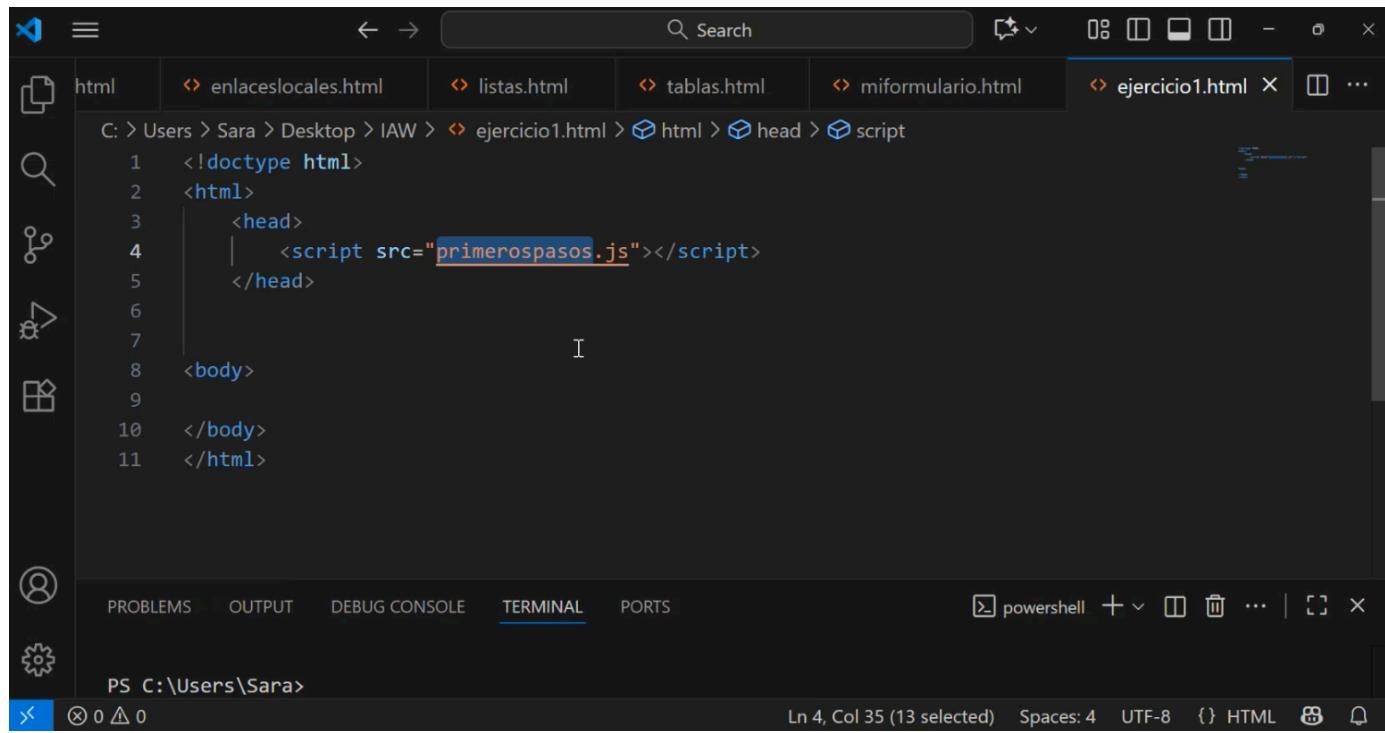
0 Desarrollo de Interfaces

Entorno de trabajo: Visual Studio Code (VSC)

Para el desarrollo de interfaces trabajaremos en **Visual Studio Code**, que será nuestro **IDE principal** (entorno de desarrollo integrado).

El trabajo con **JavaScript (JS)** se realizará en **archivos externos**, que se vinculan desde el documento HTML mediante la etiqueta:

```
<script src="primerospasos.js"></script>
```



Esto permite separar la lógica del programa del contenido HTML, mejorando la organización y mantenimiento del código.

Comentarios en JavaScript

Existen dos tipos de comentarios:

```
// Comentario en una sola línea
```

```
/* Comentario  
en bloque o multilínea */
```

Los comentarios **no se ejecutan** y sirven para documentar el código o desactivar temporalmente partes del mismo.

◆ Variables y tipos de datos

JavaScript **no es un lenguaje tipado** (a diferencia de Java o C#).

Esto significa que una misma variable puede almacenar distintos tipos de datos a lo largo del programa.

Formas de declarar variables:

```
let nombre;    // variable local  
var edad;     // variable global
```

Cláusula	Alcance	Características
let	Bloque/local	Recomendado. Optimiza recursos del sistema y evita conflictos.
var	Global	Afecta a todo el script. Puede provocar errores si se reutiliza.

Tipos de datos primitivos

Tipo	Ejemplo	Descripción
String	"Hola"	Texto entre comillas.
Number	25 , 2.34	Números enteros o decimales.
Boolean	true , false	Valores lógicos.
Undefined	variable sin valor asignado.	let x;
Null	ausencia intencionada de valor.	let y = null;
Object	{nombre: "Eva", edad:25}	Estructura con propiedades.
Array	[1,2,3,4]	Lista ordenada de valores.

◆ Reglas para nombrar variables

1. No usar **caracteres especiales**: @ # ; . ñ
2. No empezar con números.
3. Evitar **palabras reservadas** del lenguaje.
4. Utilizar nombres descriptivos: nombreUsuario , totalVenta , etc.

◆ Mostrar información desde JavaScript

Para escribir directamente en la página web se usa el método `document.write()`:

```
document.write("Texto desde JavaScript");  
document.write("El valor de numero1 es: " + numero1);  
document.write("<strong>El valor de cansado es:</strong> " + cansado + "<br>");
```

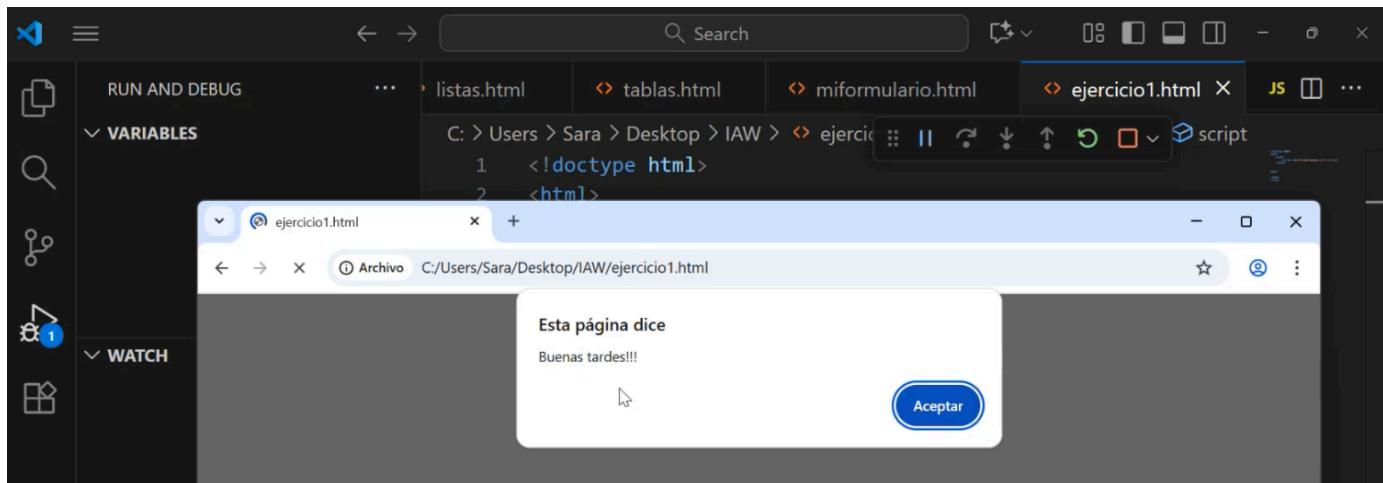
Este método permite incluir etiquetas HTML dentro del texto mostrado.

◆ Ventanas emergentes del objeto window

El objeto `window` hace referencia a la **ventana del navegador**.

Con él podemos mostrar o solicitar información al usuario:

Método	Descripción	Ejemplo
<code>alert()</code>	Muestra un mensaje.	<code>window.alert("Buenas tardes!!!");</code>
<code>prompt()</code>	Solicita un valor al usuario.	<code>window.prompt("Introduce un valor:");</code>
<code>confirm()</code>	Solicita confirmación (sí/no).	<code>window.confirm("¿Deseas continuar?");</code>



Ejemplo práctico:

```
window.alert("Buenas tardes!!!");
let edad = window.prompt("Introduce tu edad:");
```

Por defecto, el método `prompt()` devuelve valores **de tipo texto** (string).

◆ Conversión de datos

Para convertir los valores obtenidos en números:

```
parseInt() // convierte a número entero
parseFloat() // convierte a número decimal
```

Cuando el `+` se usa entre números y cadenas, JS convierte todo a texto.

Esto genera errores comunes en principiantes.

Ejemplo:

```
let num1 = 10; let num2 = "20";
let resultado = num1 + num2;
// "1020" (concatenación, no suma)
```

Para evitarlo:

```
let resultado = num1 + parseInt(num2);
// 30
```

Añadir esta nota ayuda a entender por qué es necesario usar `parseInt()`.

Ejemplo:

```
let num1 = parseInt(window.prompt("Introduce valor 1:"));
let num2 = parseInt(window.prompt("Introduce valor 2:"));
let decimal = parseFloat(window.prompt("Introduce valor decimal:"));
```

◆ Operadores aritméticos

Operador	Descripción	Ejemplo	Resultado
+	Suma	5 + 3	8
-	Resta	5 - 3	2
*	Multiplicación	5 * 3	15
/	División	10 / 2	5
%	Módulo (resto)	10 % 3	1

Ejemplo de uso:

```
let num1 = parseInt(window.prompt("Introduce valor 1:"));
let num2 = parseInt(window.prompt("Introduce valor 2:"));

let suma = num1 + num2;
window.alert("El resultado de la suma es: " + (num1 + num2));
```

- ◆ Importante: los **paréntesis** determinan la prioridad operacional.

Si no se usan, el texto se concatenará antes de realizar la suma.

◆ Operadores de asignación

Operador	Significado	Ejemplo	Resultado
=	Asignación	num1 = 25	25
+=	Suma y asigna	num1 += 10	35
-=	Resta y asigna	num1 -= 5	30
*=	Multiplica y asigna	num1 *= 2	60
/=	Divide y asigna	num1 /= 10	6
%=	Módulo y asigna	num1 %= 2	0

⚠ Los textos solo pueden **sumarse (concatenarse)** con `+`, pero **no restarse, dividirse ni multiplicarse**.

◆ Estructuras condicionales

Las estructuras condicionales permiten **ejecutar distintas acciones según una condición**.

Sintaxis general del `if`:

```
if (condición) {
    // Código si la condición se cumple
} else {
```

```
// Código si no se cumple  
}
```

💡 Ejemplo: comprobar si un número es par o impar

```
let dato = parseInt(window.prompt("Introduce un número:"));  
  
if (dato % 2 == 0) {  
    window.alert("El número es par");  
} else {  
    window.alert("El número es impar");  
}
```

◆ Operadores de comparación

Operador	Significado	Ejemplo	Resultado
==	Igualdad de valores	5 == "5"	true
!=	Diferente	5 != 3	true
>	Mayor que	5 > 3	true
<	Menor que	2 < 8	true
>=	Mayor o igual que	5 >= 5	true
<=	Menor o igual que	4 <= 5	true

◆ Ejemplo con condicionales anidados

```
let dato = parseInt(window.prompt("Introduce un valor entero:"));  
  
if (dato == 0) {  
    window.alert("El dato es igual a cero");  
} else if (dato < 0) {  
    window.alert("El dato es menor que cero");  
} else {  
    window.alert("El dato es mayor que cero");  
}
```

◆ Operadores lógicos

Operador	Descripción	Ejemplo	Resultado
&&	AND (y) — ambas condiciones deben cumplirse	(x > 0 && y > 0)	true si ambas son verdaderas
'		'	OR (o) — basta con que una se cumpla
!	NOT (no) — invierte el resultado	!(x > 0)	true si x es menor o igual a 0

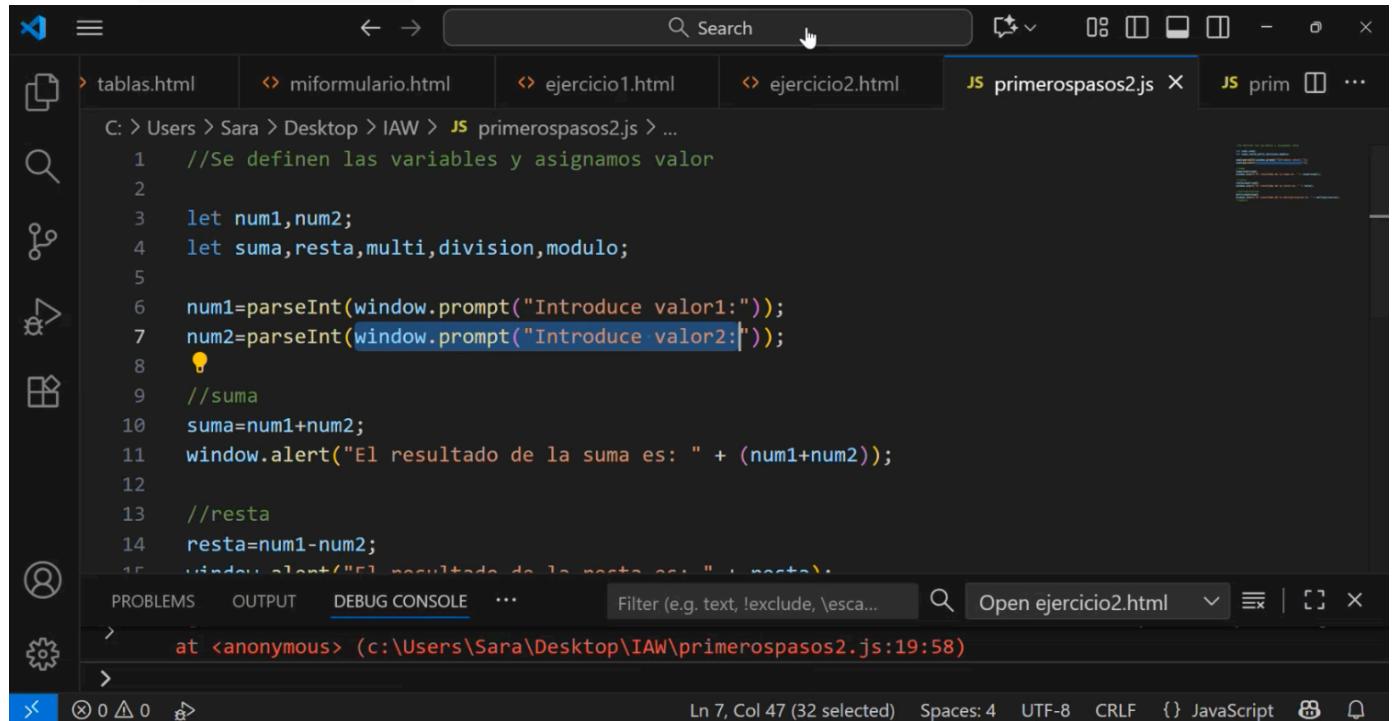
💡 Ejemplo:

```
let edad = parseInt(window.prompt("Introduce tu edad:"));
```

```
if (edad >= 18 && edad <= 65) {  
    window.alert("Edad laboral activa");  
} else {  
    window.alert("Fuera del rango laboral");  
}
```

◆ Ejemplo completo de la clase

Archivo: primerospasos2.js



```
C: > Users > Sara > Desktop > IAW > JS primerospasos2.js > ...  
1 //Se definen las variables y asignamos valor  
2  
3 let num1,num2;  
4 let suma,resta,multi,division,modulo;  
5  
6 num1=parseInt(window.prompt("Introduce valor1:"));  
7 num2=parseInt(window.prompt("Introduce valor2:"));  
8  
9 //suma  
10 suma=num1+num2;  
11 window.alert("El resultado de la suma es: " + (num1+num2));  
12  
13 //resta  
14 resta=num1-num2;  
15 window.alert("El resultado de la resta es: " + resta);
```

PROBLEMS OUTPUT DEBUG CONSOLE ... Filter (e.g. text, !exclude, \esca... Open ejercicio2.html 🔍 | ⚙️ | ✎ X

> at <anonymous> (c:\Users\Sara\Desktop\IAW\primerospasos2.js:19:58)
>

Ln 7, Col 47 (32 selected) Spaces: 4 UTF-8 CRLF {} JavaScript 🌐 📡

```
2
3 let num1,num2;
4 let suma,resta,multi,division,modulo;
5 let decimal;
6
7
8 num1=parseInt(window.prompt("Introduce valor1:"));
9 num2=parseInt(window.prompt("Introduce valor2:"));
10 decimal=parseFloat(window.prompt("Introduce valor decimal:"));
```

```
PROBLEMS OUTPUT DEBUG CONSOLE ... Filter (e.g. text, !exclude, \esca... Open ejercicio2.html x
```

```
at <anonymous> (c:\Users\Sara\Desktop\IAW\primerospasos2.js:19:58)
```

```
← → Search JS primerospasos2.js x JS primerospasos.js x
```

```
C: > Users > Sara > Desktop > IAW > JS primerospasos2.js > ...
```

```
42 //igual a cero
43 //mayor que cero
44 //menor que cero
45
46 let dato2=parseInt(window.prompt("Introduce un valor entero"));
47
48 if(dato1==0)
49 {
50     window.alert("El dato es igual a cero");
51 }
52 else if(dato1<0)
53 {
54     window.alert("El dato es menor que cero");
55 }
56 else
57 {
58     window.alert("El dato es mayor que cero");
59 }
```

```
// Se definen las variables
let num1, num2, suma, resta, multi, division, modulo, decimal;

// Solicitamos valores
num1 = parseInt(window.prompt("Introduce valor 1:"));
num2 = parseInt(window.prompt("Introduce valor 2:"));
decimal = parseFloat(window.prompt("Introduce valor decimal:"));

// Operaciones
suma = num1 + num2;
window.alert("El resultado de la suma es: " + (num1 + num2));

// Comprobación de número positivo, negativo o cero
let dato = parseInt(window.prompt("Introduce un número entero:"));

if (dato == 0) {
    window.alert("El dato es igual a cero");
} else if (dato < 0) {
    window.alert("El dato es menor que cero");
} else {
```

```
    window.alert("El dato es mayor que cero");
}
```

◆ Avance hacia la siguiente sesión

En la próxima clase se introducirán las **estructuras de repetición (bucles)**, partiendo de `for` , `while` y `do...while` , para automatizar tareas dentro del código.
